P V01 D12 20

Patentnummer: EP0807911, A3

Publikationsdatum: 1997-11-19

Uppfinnare: KALISKI BURTON S JR (US)

Sökande:: RSA DATA SECURITY INC (US)

Sökt patent: JP11003033

Ansökningsnummer: EP19970303229 19970512

Prioritetsnummer: US19960648442 19960515; US19970845196 19970421

IPC klassifikation: G07F19/00 ; G06F17/60 ; H04L9/32

EC klassifikation: G07F7/10F6, G06F1/00N5A2C, H04L9/32T

EC klassifikation: G07F7/10F6 ; G06F1/00N5A2C ; H04L9/32T

Ekvivalenter: US6085320

Utdrag

A protocol for establishing the authenticity of a client to a server in an electronic transaction by encrypting a certificate with a key known only to the client and the server. The trust of the server, if necessary, can be established by a public key protocol. The client generates and sends over a communications channel a message containing at least a part of a certificate encrypted with the server's public key or a secret session key. The server receives and processes the message to recover at least part of the certificate, verifies and accepts it as proof of the client's authenticity.

Beskrivning

The invention relates to a protocol for one party to an electronic transaction, as for example a client in a client-server transaction, to prove its authenticity to the other party of the transaction.

BACKGROUND OF THE INVENTION

Client-server systems provide electronic access by the client to data, information, accounts and other material stored at the server. In financial transactions, the system provides a client electronic access to accounts and financial resources.

In a client-server transaction, the client is required to prove to the server that it is an authentic client, and not some impersonator or other unauthorized party. Protocols are known by which a client proves to a server its authenticity, while at the same time it does not reveal information that could be misused by a third party.

A standard well known protocol for proving authenticity involves public-key cryptography. The client establishes a public key/private key pair and provides the public key to the server. In a transaction, to prove its authenticity to the server, the client forms a digital signature with its private key on a time-varying message, and the server verifies the digital signature with the client's public key. The time-varying message, which may be a timestamp or a challenge supplied by the server, is different in each instance. This message, when checked by the server, provides safeguards against a third party impersonating the client by simply replaying copies of previous signatures of the client that the third party has intercepted or otherwise acquired.

In the standard protocol described above, the server trusts that the public key belongs to the client, i.e., that the client is in fact actively involved in the transaction because it is presumed that only the client knows the private key and can form valid digital signatures. A convenient way to establish trust in a public key is to use a certificate. This is accomplished by a certification authority issuing public-key certificates signed with the certification authority's private key, which thereby asserts to the server that the client's public key is a valid public key issued by or registered with the certification authority. Assuming the server trusts the certification authority's public key, then it trusts the client's certificate, the client's public key and ultimately the client's authenticity.

With typical public-key cryptosystems, it is computationally expensive to form digital signatures because of the need to perform an exponentiation operation. In some electronic transactions, for example, those involving a smart card client where the computational capacity is limited, the standard protocol using a digital signature is computationally expensive and is therefore a significant burden.

Beller and Yacobi, in an article entitled "Fully-Fledged Two-Way Public Key Authentication and Key Agreement for Low-Cost Terminals" ELECTRONICS LETTERS, May 27, 1993, Vol. 29, No. 11, at pages 999-1000, describe a protocol that provides for less on-line computation on one side of the protocol. In this protocol authentication of the server by the client is carried out by the server sending a random challenge with an expected "colour", structure or format, to the client for verification by the client. Authentication of the client by the server is achieved by the client sending to the server its identity, public key, certificate and a signature on the random challenge for verification of the certificate and the signature by the server. The protocol is described as being useful where one side of the interaction is a low-cost customer device such as portable telephones, home banking terminals, smart cards and notebook computers.

Other protocols are known for establishing the authenticity of a client to a server. Client authentication protocols such as those based on secret-key cryptography exist, but often

and
Figure 12 illustrates a still further flow diagram of essential elements of a system having a more general protocol but which may have features of other embodiments disclosed herein added thereto.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The specific description of the invention is set forth in the environment of a smart card client. However, the invention is not limited to a smart card client since the disclosed protocols are applicable to client/server systems in general, and in particular clients having low computational capacity such as portable telephones, notebook computers and home banking terminals. In an even more general manner the disclosed exemplary embodiments as described later can involve a user, which can be an individual, a computer or some other entity, which is connected to a verifier, which can be a client, a server or some other entity, via an encrypted communications channel whereby the user can confidentially deliver to the verifier information essential to verify the message.

A smart card includes a microchip containing a processor and memories to hold programs and data. Figure 1 illustrates a smart card 1 comprising a processor 2, an erasable programmable read only memory (EPROM) 3, programmable read only memory (PROM) 4, random access memory (RAM) 5, input/output (I/O) port 6, number generator 7, clock 8 and power source 9. PROM 4 holds the card operating system and RAM 5 holds temporary results of calculations. EPROM 3 holds the certificate for the card. This certificate for the card, unlike the usual public key certificate, need not include the public key of the client since authentication of the client by the server does not rely on the public key of the client. A cache of public keys of one or more servers may also be stored in PROM 4 or EPROM 3. Number generator 7 provides random seed numbers to the processor for generating secret session keys. Clock 8, conventional and well known in the art, is used for generating a timestamp and for verifying a received timestamp. Clock 8 is optional where the server's timestamp or time-varying value is used by the client to provide a time-varying value or where a challenge procedure is followed. Power source 9 is a battery when a card has a clock. Otherwise, power may be supplied by an external source or a server. I/O terminals 6 provide a means for external communications.

The public key of a trusted certification authority may be stored in PROM 4 or EPROM 3. PROM 4, or the RAM 5 if non-volatile, may have a section for storing certificate revocation lists (CRLs). Such a list would include a list of servers whose certificates have expired or been revoked. This list would be provided by signed and dated messages from the trusted certification authority either directly or indirectly while in a communicating relationship with a server. Reference to the list during the initial stages of the protocol will indicate whether the transaction being initiated is with a valid server or with one holding a revoked certificate, and thereby whether a received server's certificate is to be verified.

The card manufacturer initializes the smart card using conventional techniques. PROM 4 is loaded with an operating program to be executed by processor 2, clock 8 is set (or an initial time-varying value, e.g., a sequence number or a timestamp is set in one of the memories when a clock is not used) and the certificate associated with the card and the trusted certification authority's public key are loaded into the memories. Optionally, server public keys and CRLs are also loaded into the memories.

A server 40 as illustrated in Figure 2 includes a processor 60, a facility for generating a time-varying value or timestamp 42, input/output port 63, and a memory 61 for holding the operating program for the processor, the private key PRIVSERV associated with the server's public key PUBSERV and the public key PUBTCA of the trusted certification authority that signed the client's certificate. In addition, memory 61 may hold a certificate revocation list (CRL) and a certificate (CERT-S) for its public key. The facility for generating a time-varying value 42 may comprise a clock for generating a timestamp or other means for generating a time-varying value. The I/O port 63 provides an interface between the processor of the server and external entities.

In the interactive embodiments of Figures 4A, 4B and 4C, message 11 consists of a certificate (CERT-S), a time-varying value, or a combination of a certificate (CERT-S) and a time-varying value. These informational items are provided to a client so that the client may properly form authenticating message 12. Figure 5 shows an interactive embodiment where the server provides a signed message 11. Since Figure 5 is the most comprehensive, it will be described first, and the embodiments of Figures 4A, 4B and 4C described primarily with respect to differentiating features caused by the differences in the content of message 11.

In Figure 5, the parties to the electronic transaction are the client 20 and the server 40. Messages 11, 12 and 13 are generated and exchanged between the client 20 and the server 40 over a communications channel 15. Successful exchanges of the messages establish the trust of the client 20 in the server 40 and the authenticity of the client 20 to the server 40. Communications channel 15 may simply be electrical connections between a card reader and the terminal equipment at a server or may be in the form of a telephone or other communications link established between a client and a remote server, or other conventional communications medium.

Client 20 includes a certificate (CERT-C) 21 stored in EPROM 3, a key generator (KEY) 22, a facility for generating a time-varying value (TS) 23 which may include the clock 8, when used, and the public key (PUBTCA) 24 of the trusted certification authority which may be stored in EPROM 3 or PROM 4. Certificate 21 comprises a message provided and signed by the trusted certification authority with its private key in the standard manner. The message in this instance need not be the client's public key because this key is not involved in the protocol. Any message is sufficient and mav be certain well structured information about the client, e.g., account number and expiration date of the account. The message may also indicate the types of transactions for which the client 20 is authorized and the period of time during which the certificate may be considered valid. The key generator 22 is comprised of any conventional means of generating an encryption key. It may comprise a subroutine in the processor and use a number supplied by a number generator 7. The facility 23 may comprise conventional clock 8 that provides a current date and time or may be one that operates on a received timestamp or time-varying value. Key 24, the public key of the trusted certification authority, is used to verify a certificate sent by the server and signed by the trusted certification authority.

Key storage unit 25 represents an optional memory or memory section for storage of keys of one or more frequently used servers. These are the public keys of servers and are made available to clients by the servers. Storing the public key of server 40 and other selected servers at the client avoids the need to process the certificate from a server to recover the key, or provides a source for the key where the certificate does not contain the public key or an easily recoverable copy of the public key of the server. CRL storage unit 26, also an optional memory or memory section, stores a list of certificates that have been revoked.

Elements 30 through 35 illustrate the functional processes of the protocol performed by the client to establish trust in the server. Public key operations are conventional well known processes in the art. Recovering the public key of the server from the server's certificate for the key and storing it in a memory, as illustrated in block 30, is a certificate processing within the skill of art.

Functional element 32 represents a public key operation performed with the trusted certification authority's public key 24 on the certificate portion of the message 11 received from the server 40. Functional element 31 represents a public key operation performed on the timestamp or other time-varying value received from a server with the server's public key obtained from processing the certificate at read and store element 30 or from key storage unit 25. At functional block 33 a standard verification procedure, as those skilled in the art appreciate, is used to verify the certificate. A certificate revocation list supplied from memory 26 may optionally be used in verifying the certificate.

Functional element 34 represents a comparison and verification of the timestamp or time-varying value received from the server 40 to verify that it is proper. Where the smart card or client has a clock, a simple comparison (allowing for small time differences) of the time at clock 8 with the time of the received timestamp suffices to verify a received timestamp. Where the smart card does not include a clock, the stored time of a last received valid

40 through a public key protocol.

Considering the Figure 5 illustration, so that client 20 may gain its trust, upon a request for access the server 40 reveals its certificate 41 by combining the certificate 41 with the signed time-varying value from facility 42 to form message 11 CERT-S{TS}PRIVSERV. Client 20 receives the message 11 sent over the communications channel 15, verifies the signature on the certificate with the trusted certification authority's public key 24 in public key operation 32 and verification process 33, and processes the certificate in operation 30 to read and store the public key of the server. Client 20 then uses the public key of the server in public key operation 31 to obtain the time-varying value. As indicated above, the public key of server 40 alternatively may be retrieved from public key storage unit 25 where used. Checking of the time-varying value to see that it is valid is done in comparison unit 34. A failure to verify the server's certificate or validate the received time-varying value terminates the transaction. Signature verification, particularly for RSA, is computationally inexpensive, so the computational burden on the client 20 is minimal. As an additional step in certificate verification, the client 20 may check the values of one or more fields of the certificate 41 to determine whether the server 40 is authorized for transactions with the client 20. It is presumed that the trusted certification authority of interest issues authorized certificates only to trusted servers, so the pair of signature verifications is sufficient for the client 20 to gain trust in the server 40.

Once client 20 verifies the time-varying value from 42 and the certificate 41 of server 40, trust of the server 40 is established. Thereafter, client 20 generates a random secret session key (KSS) at 22, combines this key with its time-varying value (TS) generated by a clock at 23, or where no clock is present replicates the time-varying value received from the server, to form a message and encrypts the message with the public key (PUBSERV) of server 40 obtained from storage at element 30 or 25 to form the encrypted message 12. Again, for RSA, encrypting with the server's public key is computationally inexpensive so this is not a burden on the client.

The encrypted message 12, {KSS;TS}PUBSERV, is sent to the server 40 where it is received and processed for recovery of the secret session key KSS by decrypting the message 12 with private key 43 in private key operation 51. A checking of the time-varying value TS demonstrates to the server 40 that a client is active in the transaction, not an impersonator replaying a recorded message.

Client 20 then encrypts its certificate 21 using the secret session key to produce encrypted message 13, {CERT-C}KSS, and sends it to server 40. Messages 12 and 13 may be combined as one message {KSS;TS}PUBSERV{CERT-C} KSS. Server 40 receives the encrypted message 13 and decrypts it in decryption operation 54 to gain the client's 20 certificate 21. Certificate 21 is processed in public key operation 55 with the trusted certification authority's public key stored at 44. After verification at 56, with or without the optional CRL in unit 45, the authenticity of client 20 is accepted by server 40 and the transaction can be undertaken.

When clocks are used for both timestamp facilities, the client and the server need to account for variations in their clocks when checking that the timestamp received is current. One procedure is to determine the difference between the two clock timestamps, for example, the client determines the difference between the received timestamp and its own clock generated timestamp, and compares that difference to a pre-set reference value to see that it is less than the reference value. Other techniques known to those skilled in the art may be used to account for the clock variations. See, for example, Weiss, U.S. Patent 4,885,778, entitled "Method and Apparatus for Synchronizing Generation of Separate Free Running, Time Dependent Equipment," which describes a technique for synchronizing client and server clocks in an authorization protocol.

In place of a timestamp a challenge may be used. A challenge may comprise any-time varying message that can be processed and verified.

The client may also store a CRL of servers. Either as a part of the authentication protocol or subsequent thereto, the server and the client may exchange lists of revoked certificates.

described embodiments. Server 40 decrypts the message at 51 with its private key, verifies at 52 the time-varying value, and processes the client's certificate as hereinbefore described to authenticate it. A smart card using this protocol may only comprise a processor, a memory storing the certificate for the card and the public key of servers and a facility for providing a time-varying value. The smart card may then engage in a non-interactive protocol with a server for the purpose of establishing its authenticity to a server. In an interactive protocol, the server in a communication to the client provides a time-varying value and its public key certificate as in Figure 4B, and the smart card processor processes the receipt of same to produce a time-varying value and the public key of the server. The other interactive embodiments of Figures 4A and 4C and Figure 5 may also be modified to have no session key.

It is also possible to modify the aforementioned embodiments to encrypt only part of the certificate, which may lead to greater efficiency in the protocol. For instance, the client can encrypt only the signature on the certificate, transmitting the rest of the certificate unencrypted. Since a certificate is not valid without a signature, an opponent who obtains only the non-signature part of the certificate will not be able to impersonate the client.

As a generalization, the client can encrypt any data essential to the verification of the certificate. The signature is one example; another example is a part of the signature, large enough so that the opponent cannot guess it. A third example is a secret certificate serial number assigned by the certification authority. In general, the most efficient approach, in terms of communication requirements, is to encrypt something that is already required by the server, rather than something new. Since the signature is already required, it is a natural choice, though other parts of the certificate may be appropriate as well.

In some cases, it may be more efficient in terms of communication bandwidth to encrypt more than just the data required to verify the certificate. For instance, if the encryption is performed with the public key of the server, then the client can encrypt as much additional data as can be encrypted with a single public-key encryption operation. The approach of encrypting the entire certificate is an extreme example. As another variation, it is possible to encrypt part of the certificate with a public-key encryption, and part with a secret-key encryption.

Figure 8 illustrates in part a variation of the embodiment of Figure 5 wherein only a portion of the certificate 21 is encrypted and transmitted to the server 40 with the remainder of the certificate (REST-C) being transmitted in unencrypted form. Thus, the certificate 21 can be split at 81 so that any data in the certificate which is essential to the verification of the certificate is split and subsequently encrypted by secret session key (KSS) at 38. The remainder of the certificate (REST-C) is transmitted unencrypted to the server 40.

Upon receipt of the transmitted encrypted and nonencrypted portions of the certificate the server 40 decrypts the former at 54 and joins the latter at element 82 so as to obtain the client's certificate 21. Thereafter as illustrated in Figure 5, the certificate is processed in public key operation 55 with subsequent verification at 56. As previously noted, the encrypted portion (X-C) of the client's certificate may be any portion which is sensitive or essential to the verification of the certificate such as the signature or a portion of the signature.

Figure 9 is an illustration of a variation of the embodiment of Figure 6 wherein only a portion (X-C) of the client's certificate 21 is encrypted with the remainder (REST-C) being transmitted in unencrypted form. In this regard the certificate 21 is again split at 81 to obtain an essential portion (X-C) of the certificate. Thereafter, as may be seen from Figures 9 and 6, the essential portion of the certificate is concatenated at 36, encrypted at 38 and transmitted to the server 40. Moreover, the unencrypted portion (REST-C) of the certificate 21 is transmitted to the server 40 for joining at 82 with the decrypted portion (X-C) for subsequent verification of the client's certificate 21.

Figure 10 illustrates a variation of Figure 7 wherein only an essential portion of the client's certificate 21 is encrypted with the remainder of the certificate being transmitted unencrypted. More specifically, the certificate 21 is split at 81 to form an essential portion (X-C) which is concatenated and encrypted at 36 and 37, respectively, for transmission to the

reader) is trusted not to reveal account numbers or a personal identification number (PIN). Thus, it is reasonable to assume that the server or card reader will not reveal certificates.

In the case of RSA, the encryption and signature operations can follow the techniques described in PKCS #1: RSA ENCRYPTION STANDARD (RSA Laboratories, November 1993), in International Standard 9796: Information Technology, Security Techniques: Digital Signature Scheme Giving Message Recovery (ISO/IEC, 1991), in M. Bellare and P. Rogaway, "Optimal Asymmetric Encryption" in Advances in Cryptology - Eurocrypt '94, pp 92-111 (Springer-Verlag (New York 1995)) or in similar standards, as are well known to those familiar with RSA.

The client and server certificates may be signed by the same trusted certification authority or different trusted certification authorities. The client and server, in either case, needs to have in its possession, the public key of the trusted certification authority that signed the received certificate in order to verify it.

In the above noted embodiments where the server's public key is provided to the client, the same server public key was used for encryption and verification. However, the public key of the server 40 for encryption may be different than the public key for verifying signatures. In Figure 5, for example, elements 31 and 37 could employ different public keys by storing two different public keys associated with the same server; one of which would be for verifying the server signatures at 31 and the other for encrypting data at 37 to be sent to the server. Alternatively, the certificate of the server could contain two separate keys along with identification as to their purposes. Moreover, in either alternative, element 43 of Figure 5, for example, would provide appropriately different private keys to elements 17 and 51.

As an additional modification to the disclosed exemplary embodiments, the client's certificate (CERT-C) may be generated once by the certification authority and stored in the client's memory 21 or it may be a certificate generated by a certification authority whenever the client is authenticated to the certification authority, e.g. as part of a daily log-in procedure. Moreover, the authentication operation could be carried out by techniques described herein or by other authentication techniques. The new certificate could also contain the time at which authentication occurred and could expire later at some set time. Thus, the exposure time of the certificate would be limited if it is obtained by an opponent. The new certificate could also specify the types of operations for which the client is currently authenticated. Under such circumstances, the client would present the new certificate to the server to authenticate itself to the server, and the server would check that the certificate has not expired and that the client is authorized for a particular type of operation.

In financial applications, a certificate can be easily authenticated since it carries the digital signature of a certification authority. An account number cannot be easily authenticated because checking is done through accessing an on-line database. Therefore, in a financial application the certificate has clear benefits over an account number or an account number in combination with a PIN verification procedure.

In addition, if the account number contains check digits, they can usually be constructed by any third party with a public algorithm. Thus a third party can easily forge account numbers. For this reason, a database check is essential. Moreover, if the check digits are computed based on a secret key stored in the server, the same secret key must be stored in all servers. Therefore, an opponent who compromises one server can forge account numbers. This is another reason for the practice of having a central database perform the check. With certificates, the server stores only the trusted certification authority's public key, not the private key. Thus an opponent that compromises a server may obtain access to certificates known to that server, but does not gain the ability to form new ones.

As previously noted, the system as illustrated in Figure 12 is more fundamental and has a more general protocol whereby a user is enabled to confidentially deliver a credential authorizing the user to perform an operation. In order to reflect the more fundamental system and protocol the terms "user", "credential" and "verifier" are used rather than "client", "certificate" and "server", respectively, so as to indicate the more general nature of the Figure 12 exemplary embodiment. In Figure 12 the credential includes information essential to verify the credential which is transmitted to a verifier by way of an encrypted

In any event, the system as illustrated in Figure 12, for example, provides the fundamental features of allowing a user to confidentially deliver to a verifier information which is essential for verifying a credential assigned by a certification authority which authorizes the user to conduct some transaction wherein the credential may or may not involve the use of a one way function but always contains the digital signature of the credential issuer.

Moreover, although the system illustrated in Figure 12 merely illustrates the functional elements and blocks of a more fundamental system involving confidential delivery to a verifier of information essential for verifying whether a user is authorized to perform an operation, various features of the previously disclosed embodiments may also be included in the Figure 12 embodiment. For example, as previously noted, encryption on the encrypted communication channel 66 may be obtained with a shared secret key pre-installed with the user and verifier or established by encryption with the verifier's public key. Alternatively, other well known techniques for establishing a secret key by agreement can be used such as through the use of a Diffie-Hellman algorithm. Additionally, encryption as in the embodiments of Figures 4A through 7 may be obtained through the use of the verifier's public key or the use of a non-repeating value such as a time stamp. Moreover, as aforementioned, the entire credential may be encrypted or only essential data of the credential may be encrypted with the remainder of the credential being transmitted unencrypted in the manners illustrated in Figures 8 through 10, for example.

While the invention has been described in connection with what is presently considered to be the most practical and preferred embodiment, it is to be understood that the invention is not to be limited to the disclosed embodiment, but on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

---

1. A method for establishing the authenticity of a client in a client-server electronic transaction where a server having a public key/private key pair and a certification authority's public key and a client having a certificate signed by the certification authority are coupled by a communications channel comprising, the client,

a) generating a secret session key,
b) producing a time-varying value,
c) concatenating one of the secret session key or the certificate with the time-varying value,
d) encrypting the secret session key or the secret session key concatenated with said time-varying value with the server's public key to produce a first message,
e) encrypting at least a part of the certificate or the certificate concatenated with the time-varying value with said secret session key to produce a second message, and
f) sending the first and second messages to the server, where a decrypting operation by the server on the first message with the private key of the public key/private key pair provides the server with the secret session key, a decrypting operation on the second message with the secret session key provides the server with at least a part of the client's certificate, the decrypting operation with one of the keys provides the time-varying value, and a verifying operation on the time-varying value and a public key operation with the certification authority's public key and verifying operation on the client's certificate establish the authenticity of the client.

2. A method as in claim 1 wherein the client's certificate comprises structural information about the client but no public key.

3. A method as in claim 2 further including said server sending a time-varying value to the client, said client using the received time-varying value for the time varying-value that is concatenated with the secret session key or the certificate.

4. A method as in claim 1 further including the server sending a time-varying value to the client, said client using the received time-varying value for the time-varying value that is concatenated with the secret session key or the certificate.

5. A method as in claim 1 wherein establishing the authenticity of the client is non-interactive.

6. A method as in claim 2 wherein establishing the authenticity of the client is non-interactive.

7. A method as in claim 1 wherein the certificate includes a first data portion which is essential for verification of the certificate and a second data portion, said method prior to executing step e) splitting said first data portion from said second data portion, sending said second data portion to the server without encryption, encrypting and decrypting the second message in steps e) and f), respectively, only said first data portion, of the certificate, joining said first and second data portions at the server and thereafter performing the verifying operation of step f).

8. A method as in claim 1, wherein the certificate includes data essential for verification of the certificate as well as non-essential data and only the essential data is encrypted and decrypted respectively in steps e) and f).

9. A method as in claim 4, wherein the certificate includes data essential for verification of the certificate as well as non-essential data and only the essential data is encrypted and decrypted respectively in steps e) and f).

10. A method for establishing the authenticity of a client in a client-server electronic transaction where a server and a client are coupled by a communications channel, comprising,

establishing a client's trust in the server over the communications channel,
the client having stored in a memory a certificate and the server's public key, and

(a) generating in a processor, upon having established the trust in the server, a secret session key, encrypting the secret session key with the server's public key and sending the encrypted secret session key as a message to the server over the communications channel, and
(b) encrypting at least a part of said certificate with said secret session key and sending the results of the encryption to the server over the communications channel,

23. A method as in claim 18 including using a clock for generating the time-varying value in the form of a timestamp.

24. A method as in claim 18 including generating the time-varying value by using the time-varying value received from the server.

25. A method as in claim 10 wherein the generating of a secret session key includes the step of using a random number generated from a random number generator.

26. A method as in claim 10 wherein the certificate includes a first data portion which is essential for verification of the certificate and a second data portion, said method prior to executing step a) splitting said first data portion from said second data portion, sending said second data portion to the server without encryption, encrypting in step a) and decrypting at the server only said first data portion, and joining said first and second data portions at the server after decrypting said results.

27. A method as in claim 10 wherein the certificate includes data essential for verification of the certificate as well as non-essential data and only the essential data is encrypted and decrypted respectively by the client and server.

28. A smart card comprising a processor, a read only memory and an input/output port,

said processor including

means for producing a trusted server's public key, and
means for executing a protocol stored in said memory for establishing the authenticity of the smart card to a trusted server and for generating a secret session key,

said read only memory having stored therein a certificate for the smart card,
said protocol including encrypting the secret session key with the trusted server's public key and sending the encrypted secret session key to the trusted server over the input/output port,
said protocol further including encrypting at least a part of said certificate for the smart card with said secret session key and sending the encrypted certificate to the trusted server over the input/output port.

29. A smart card as in claim 28 wherein said read only memory has stored therein a plurality of public keys belonging to respectively different servers which may be used as a source from which the trusted sewers' public key is produced.

30. A smart card as in claim 28 further comprising a facility for providing a time-varying value and means for concatenating the time-varying value with the secret session key or the certificate.

31. A smart card as in claim 30 wherein said facility for providing a time-varying value comprises a clock.

32. A smart card as in claim 30 wherein said facility for providing a time-varying value comprises means for storing a received time-varying value from a server and for supplying it or a modification thereof as the time-varying value of the smart card.

33. A smart card as in claim 30 wherein said processor further comprises means for establishing trust in a server, said means including using a public key operation with a certification authority's public key for establishing the identity of a server and a timestamp of the facility for providing a time-varying value for establishing that a message received from a server comprising a timestamp is current.

34. A smart card as in claim 28 wherein said certificate for the smart card includes structured information about the smart card but no public key.

35. A smart card as in claim 28 wherein said means for producing a trusted server's public key includes processing a public key certificate received from a server.

36. A smart card as in claim 34 further comprising a timestamp facility providing timestamps for verifying received timestamps and supplying a timestamp to messages generated by the smart card.

37. A smart card as in claim 28 further comprising

said memory having stored therein a certificate for the smart card,
said protocol including concatenating at least a part of the certificate with a time-varying value provided by said facility, encrypting the result of the concatenating with the server's public key to form a message and sending the message via the input/output port.

45. A smart card as in claim 44 wherein said facility for providing a time-varying value comprises a clock.

46. A smart card as in claim 44 wherein the means for producing a server's public key includes receiving, processing and verifying a public key certificate of the server provided at the input/output port during a transaction.

47. A smart card as in claim 46 wherein the server's public key used for verifying and encrypting are different public keys of the server.

48. A smart card as in claim 44 wherein said facility for providing a time-varying value comprises means for receiving a time-varying value via the input/output port and for supplying it or a modification thereof as the time varying value of the smart card.

49. A smart card as in claim 48 wherein the means for producing a server's public key includes receiving and processing and verifying a public key certificate received from a server at the input/output port.

50. A smart card as in claim 44 wherein said certificate for the smart card includes structured information about the smart card but no public key.

51. A smart card as in claim 44 wherein the certificate includes a first data portion and a second data portion and said protocol concatenates and encrypts only the first data portion and the message sent via the input/output port includes an encrypted first data portion and an unencrypted second portion.

52. A method for establishing the authenticity of a client in a client-server electronic transaction where a server having public key/private key pairs and a certification authority's public key and a client having a certificate signed by the certification authority are coupled by a communications channel comprising

the client,
producing a time-varying value, concatenating the time-varying value with at least part of said certificate, encrypting the concatenated result with the server's public key to produce a message, and sending the message to the server,
the server,
decrypting the message using the private key of the public key/private key pair recovers the time-varying value and at least part of the client's certificate, verifies the time-varying value and processes at least part of the certificate using the certification authority's public key and verifies it, establishing the authenticity of the client.

53. A method as in claim 52 wherein the certificate includes a first data portion and a second data portion and only the first data portion is concatenated and encrypted and the message sent to the server includes an encrypted first data portion and an unencrypted second data portion, said server decrypting the first data portion and joining the decrypted first data portion and the second data portion.

54. A method as in claim 53 wherein the first data portion of the certificate comprises at least a portion of a digital signature on the certificate.

55. A method as in claim 52 wherein the client's certificate comprises structural information about the client but no public key.

56. A method as in claim 55 wherein said server has a public key certificate for its public key and steps in the method include sending said public key certificate to the client, said client processing the public key certificate to verify it and to recover the server's public key and using the public key in the step of encrypting the concatenated result.

57. A method as in claim 56 further including said server sending a time-varying value to the client, said client using the received time-varying value for the time-varying value that is concatenated with at least

a non-repeating value is included with the data essential for verification.

74. A method as in claim 65 wherein the communication channel is encrypted with the verifier's public key and a non-repeating value is included with the data essential for verification.

75. A method as in claim 65 wherein the verifier's certificate is provided to the user and the user verifies the certificate.

76. A method as in claim 65 wherein the entire credential is transmitted to the verifier through the encrypted communications channel.

77. A method as in claim 65 wherein the data essential to verify the credential includes at least the digital signature.

78. A method as in claim 65 wherein the data essential to verify the credential includes at least a secret value whose one-way function value is contained in the credential.

79. A method as in claim 65 wherein the data essential to verify the credential includes at least a path through a hash tree whose root is contained in the credential.

80. A method as in claim 77 wherein the verifier checks the digital signature using the credential issuing authority's public key.

81. A method as in claim 78 wherein the verifier computes and compares the one-way function of the secret value with the one-way function value of the credential.

82. A method as in claim 79 wherein the verifier checks the path through the hash tree using the root contained in the credential.

83. A method as in claim 66 wherein the verifier checks the credential using the credential issuing authority's public key.

84. A method as in claim 65 wherein the credential is issued as a result of successful authentication to the credential issuing authority.

85. A method as in claim 65 wherein the credential issuing authority is the verifier.

86. A method as in claim 84 wherein the credential issued includes a field computed as a one-way function of a secret value, the secret value is not revealed to the credential issuing authority and the secret value is transmitted as essential data.

87. A method as in claim 84 wherein the credential issued includes a root of a hash tree and a path through the hash tree is transmitted as essential data through the encrypted communication channel and wherein one or more leaves of the hash tree are not revealed to the credential issuing authority.

88. A method as in claim 84 wherein the determination of authorization includes a determination that the credential is not included in a credential revocation list.

89. A method as in claim 65 wherein the credential includes data specifying the types of operations for which the user is authorized, or the time at which the credential expires or a list of authorized verifiers.

90. A method as in claim 65 wherein the credential includes the user's public key but the user's public key is not used in the determination of authorization.

91. A system for determining whether a user is authorized to perform an operation with a verifier, said system comprising:

a user element for providing all or at least a part of the data included in a credential, said user element also providing data essential to verify the credential, said credential including at least a digital signature by a credential issuing authority;
said user element having a protocol for engaging with the verifier and including a device which selects data for transmission to the verifier, said data selected by the device including at least data essential to

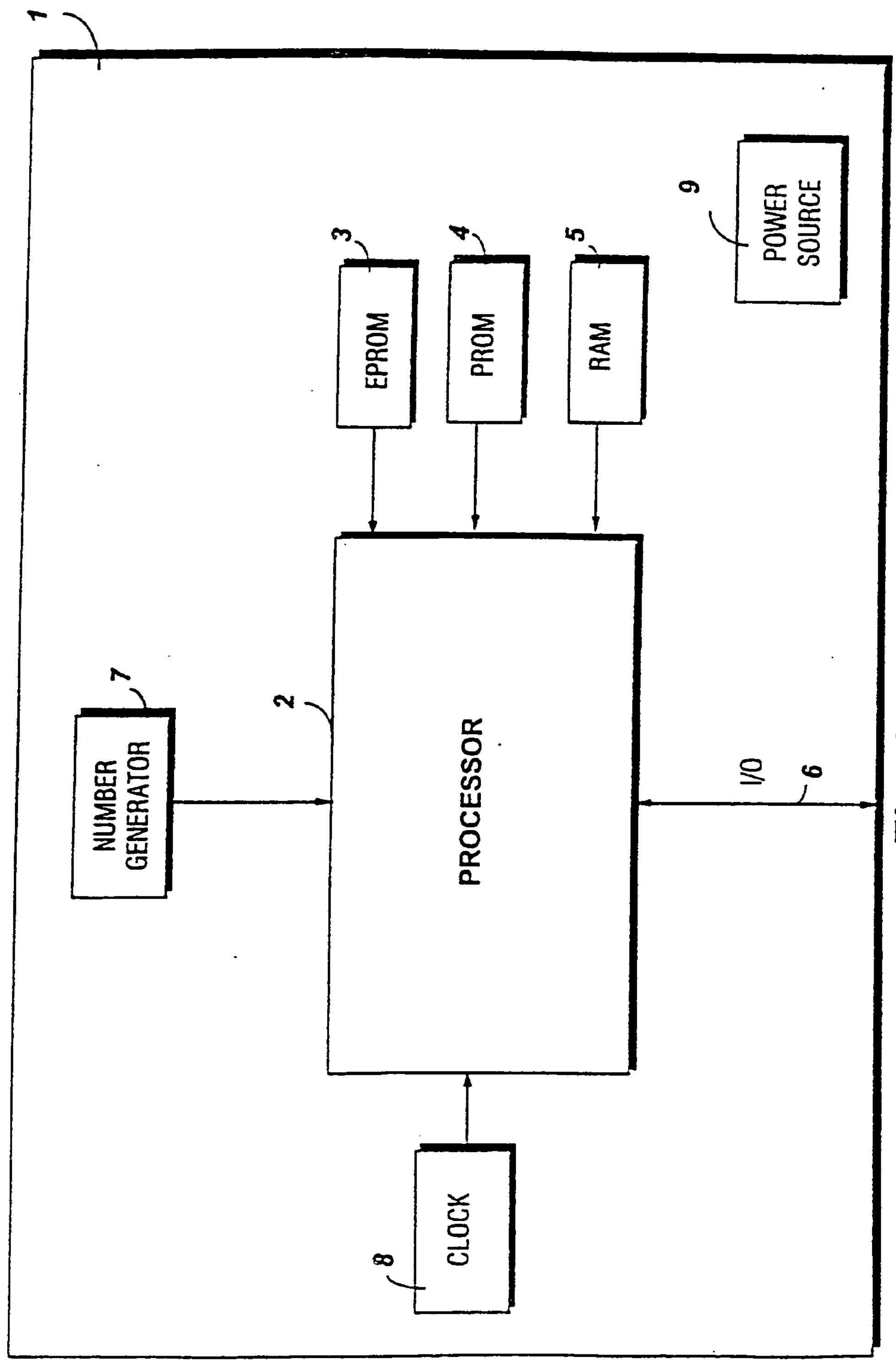

1
2
PROCESSOR
3
EPROM
4
PROM
5
RAM
9
POWER SOURCE
7
NUMBER GENERATOR
8
CLOCK
I/O
6

Fig. 1

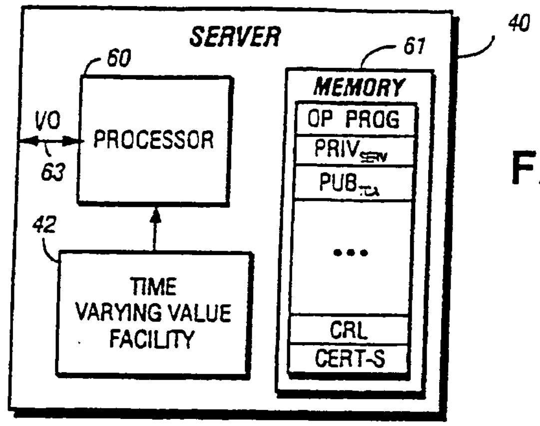

SERVER
40
60
61
MEMORY
I/O
PROCESSOR
63
OP PROG
PRIV SERV
PUB TCA
42
TIME VARYING VALUE FACILITY
CRL
CERT-S

Fig.2

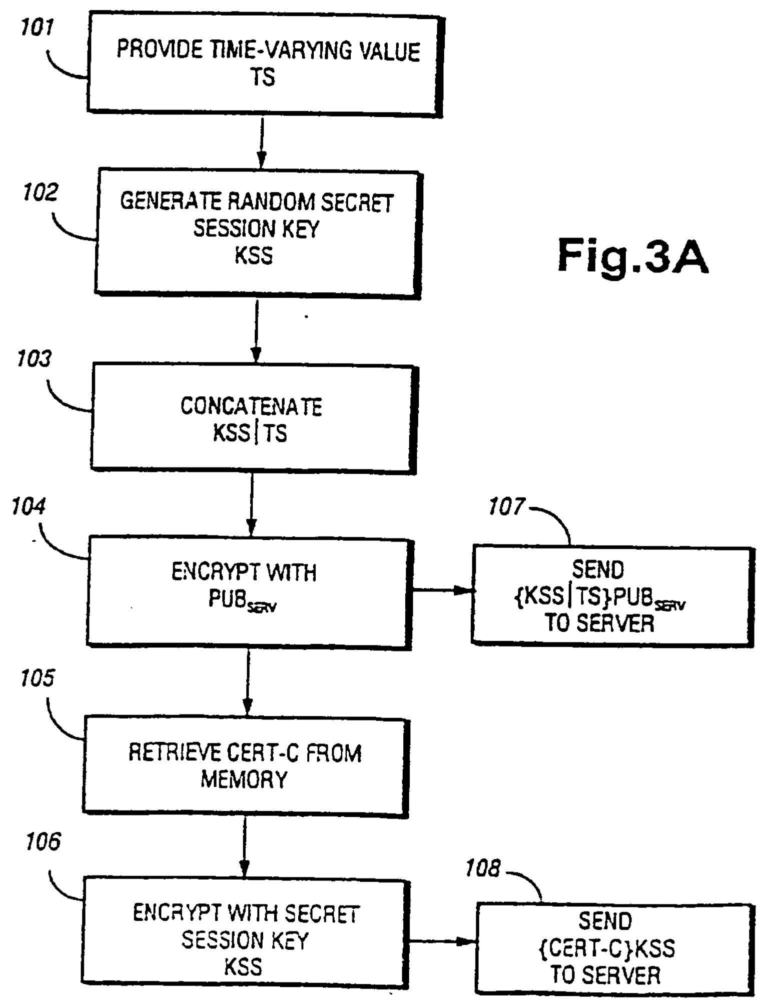
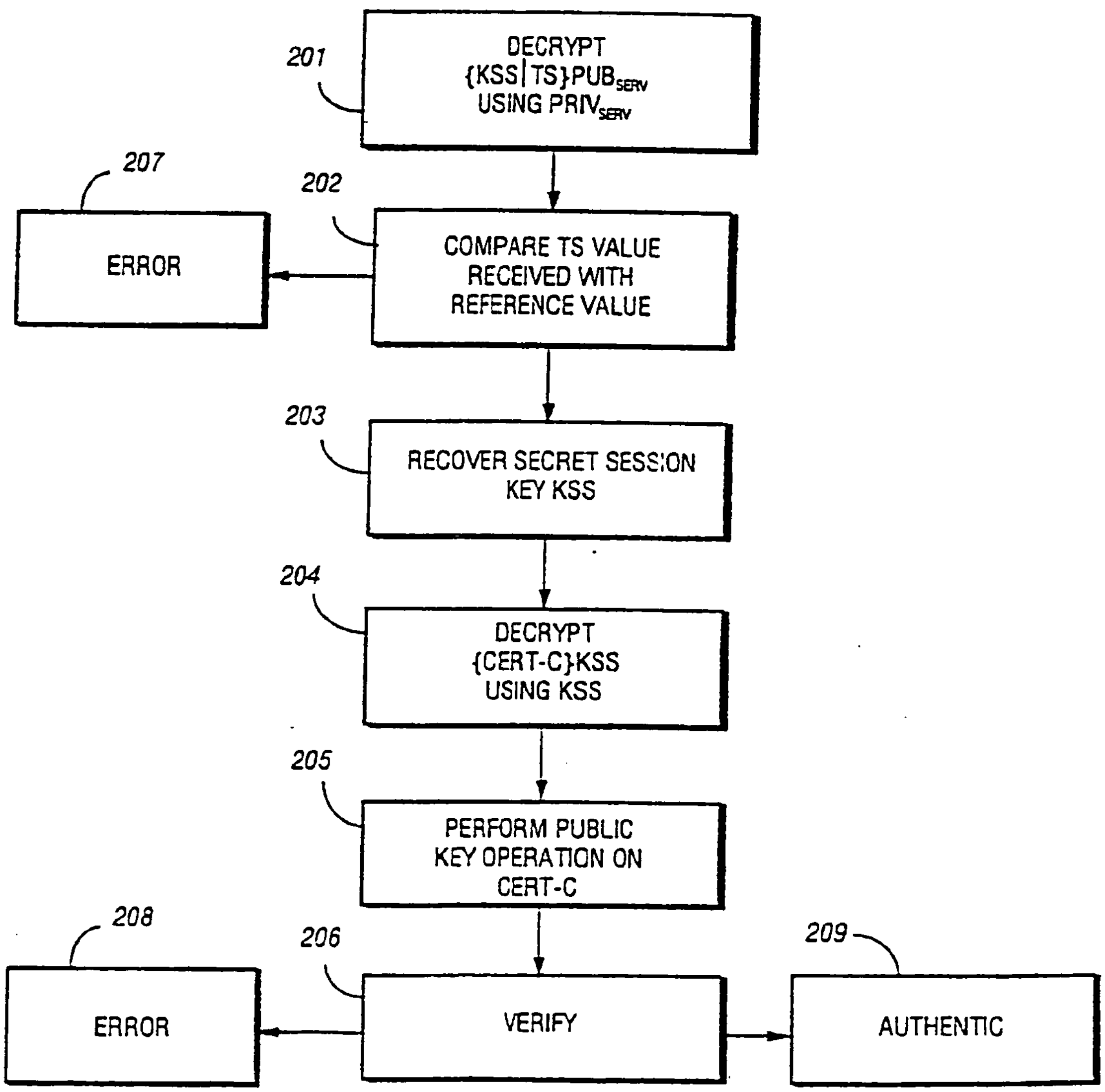

101
PROVIDE TIME-VARYING VALUE TS
102
GENERATE RANDOM SECRET SESSION KEY KSS
103
CONCATENATE KSS|TS
104
ENCRYPT WITH PUB SERV
107
SEND {KSS|TS}PUB SERV TO SERVER
105
RETRIEVE CERT-C FROM MEMORY
106
ENCRYPT WITH SECRET SESSION KEY KSS
108
SEND {CERT-C}KSS TO SERVER

Fig.3A

Fig.3B


DECRYPT
{KSS|TS}PUB SERV
USING PRIV SERV
201
COMPARE TS VALUE
RECEIVED WITH
REFERENCE VALUE
202
ERROR
207
RECOVER SECRET SESSION
KEY KSS
203
DECRYPT
{CERT-C}KSS
USING KSS
204
PERFORM PUBLIC
KEY OPERATION ON
CERT-C
205
VERIFY
206
ERROR
208
AUTHENTIC
209

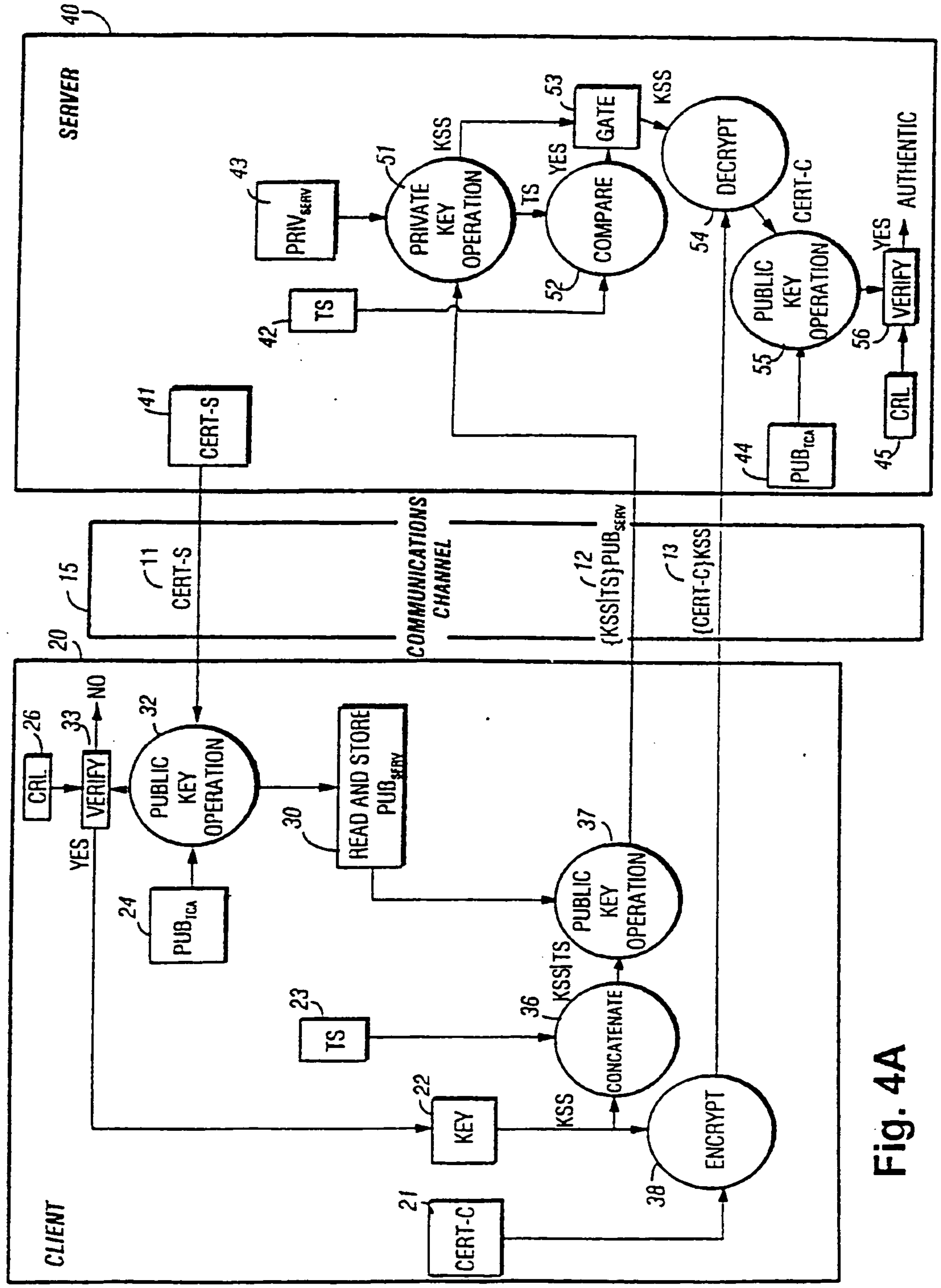
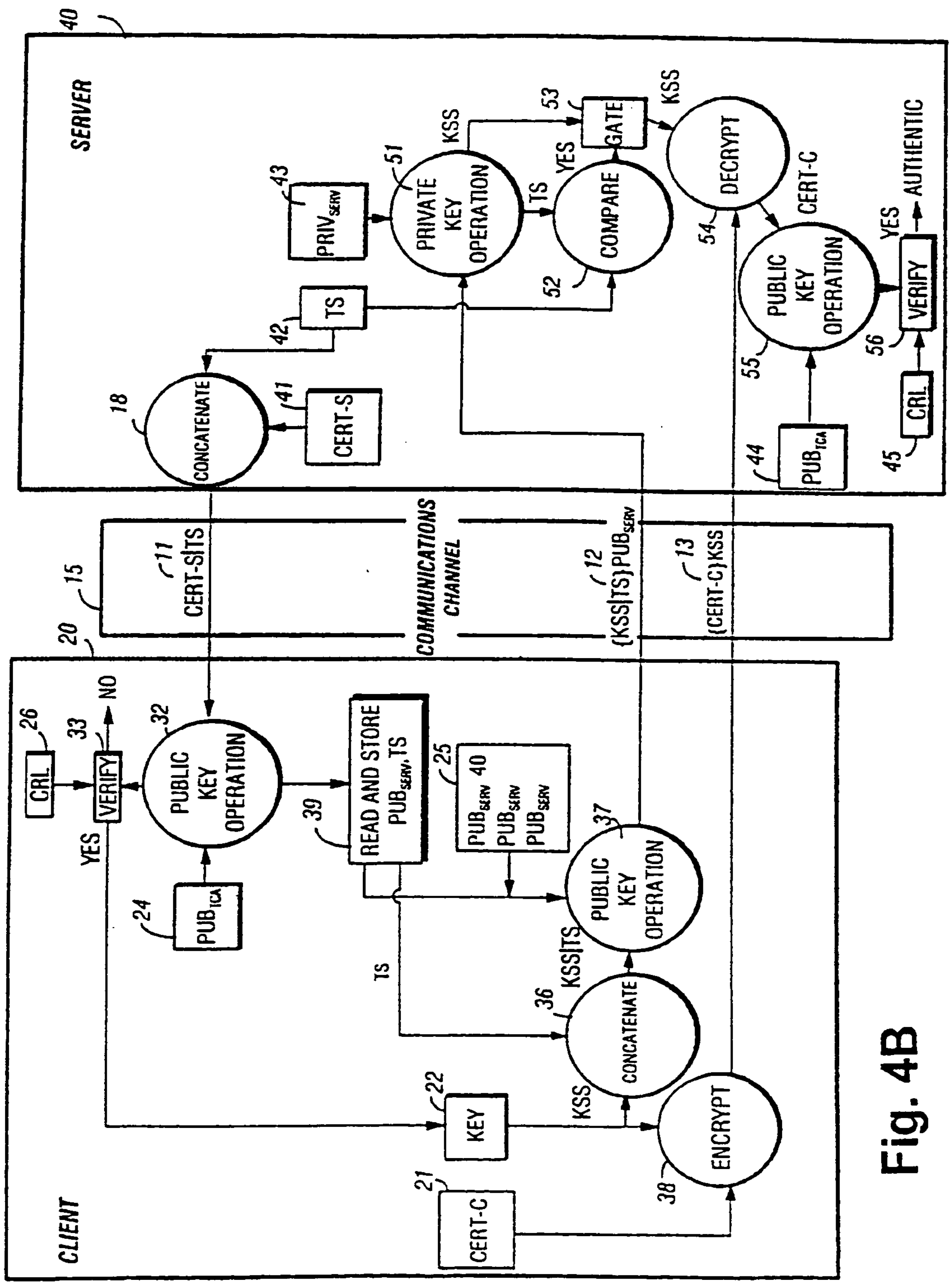

SERVER
40
CERT-S
41
TS
42
PRIV_SERV
43
PUB_TCA
44
CRL
45
PRIVATE KEY OPERATION
51
KSS
TS
COMPARE
52
YES
GATE
53
KSS
DECRYPT
54
CERT-C
PUBLIC KEY OPERATION
55
VERIFY
56
YES
AUTHENTIC
COMMUNICATIONS CHANNEL
15
CERT-S
11
{KSS|TS}PUB_SERV
12
{CERT-C}KSS
13
CLIENT
20
CERT-C
21
KEY
22
TS
23
PUB_TCA
24
CRL
26
VERIFY
33
NO
YES
PUBLIC KEY OPERATION
32
READ AND STORE PUB_SERV
30
CONCATENATE
36
KSS|TS
PUBLIC KEY OPERATION
37
KSS
ENCRYPT
38

Fig. 4A


SERVER
CLIENT
40
20
15
COMMUNICATIONS CHANNEL
11 CERT-S|TS
12 {KSS|TS}PUB$_{SERV}$
13 {CERT-C}KSS
18 CONCATENATE
41 CERT-S
42 TS
43 PRIV$_{SERV}$
51 PRIVATE KEY OPERATION
KSS
TS
52 COMPARE
YES
53 GATE
KSS
54 DECRYPT
CERT-C
55 PUBLIC KEY OPERATION
56 VERIFY
YES
AUTHENTIC
44 PUB$_{TCA}$
45 CRL
26 CRL
33 VERIFY
NO
YES
32 PUBLIC KEY OPERATION
24 PUB$_{TCA}$
39 READ AND STORE PUB$_{SERV}$, TS
25 PUB$_{SERV}$ 40 PUB$_{SERV}$ PUB$_{SERV}$
37 PUBLIC KEY OPERATION
KSS|TS
TS
36 CONCATENATE
KSS
22 KEY
38 ENCRYPT
21 CERT-C
Fig. 4B

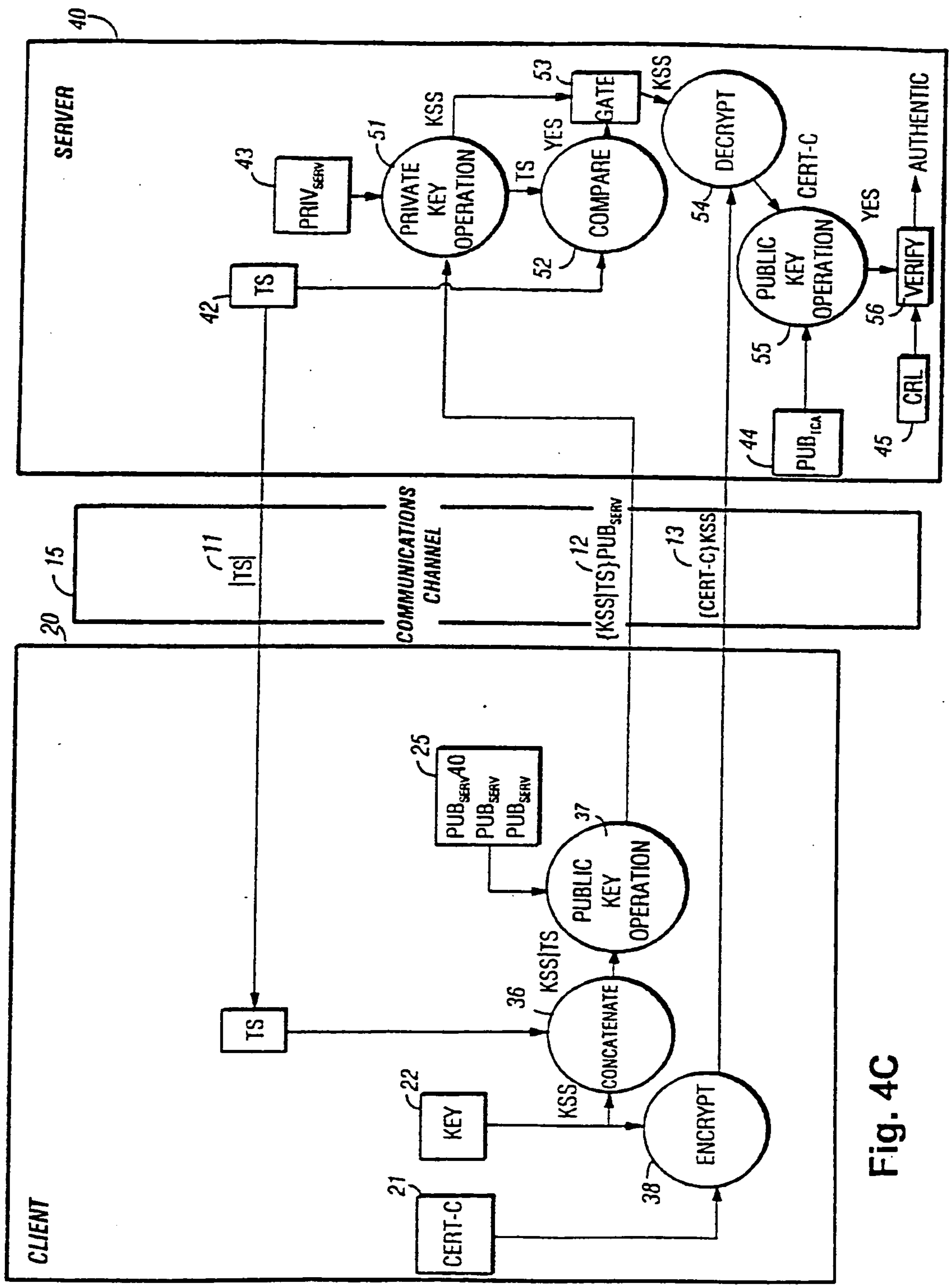

SERVER
40
43
PRIV SERV
51
PRIVATE KEY OPERATION
KSS
53
GATE
KSS
DECRYPT
TS
YES
COMPARE
52
54
CERT-C
42
TS
PUBLIC KEY OPERATION
YES
AUTHENTIC
55
56
VERIFY
44
PUB TCA
45
CRL
COMMUNICATIONS CHANNEL
15
11
|TS|
12
{KSS|TS}PUB SERV
13
{CERT-C}KSS
20
CLIENT
25
PUB SERV 40
PUB SERV
PUB SERV
37
PUBLIC KEY OPERATION
KSS|TS
36
CONCATENATE
TS
22
KEY
KSS
ENCRYPT
38
21
CERT-C

Fig. 4C

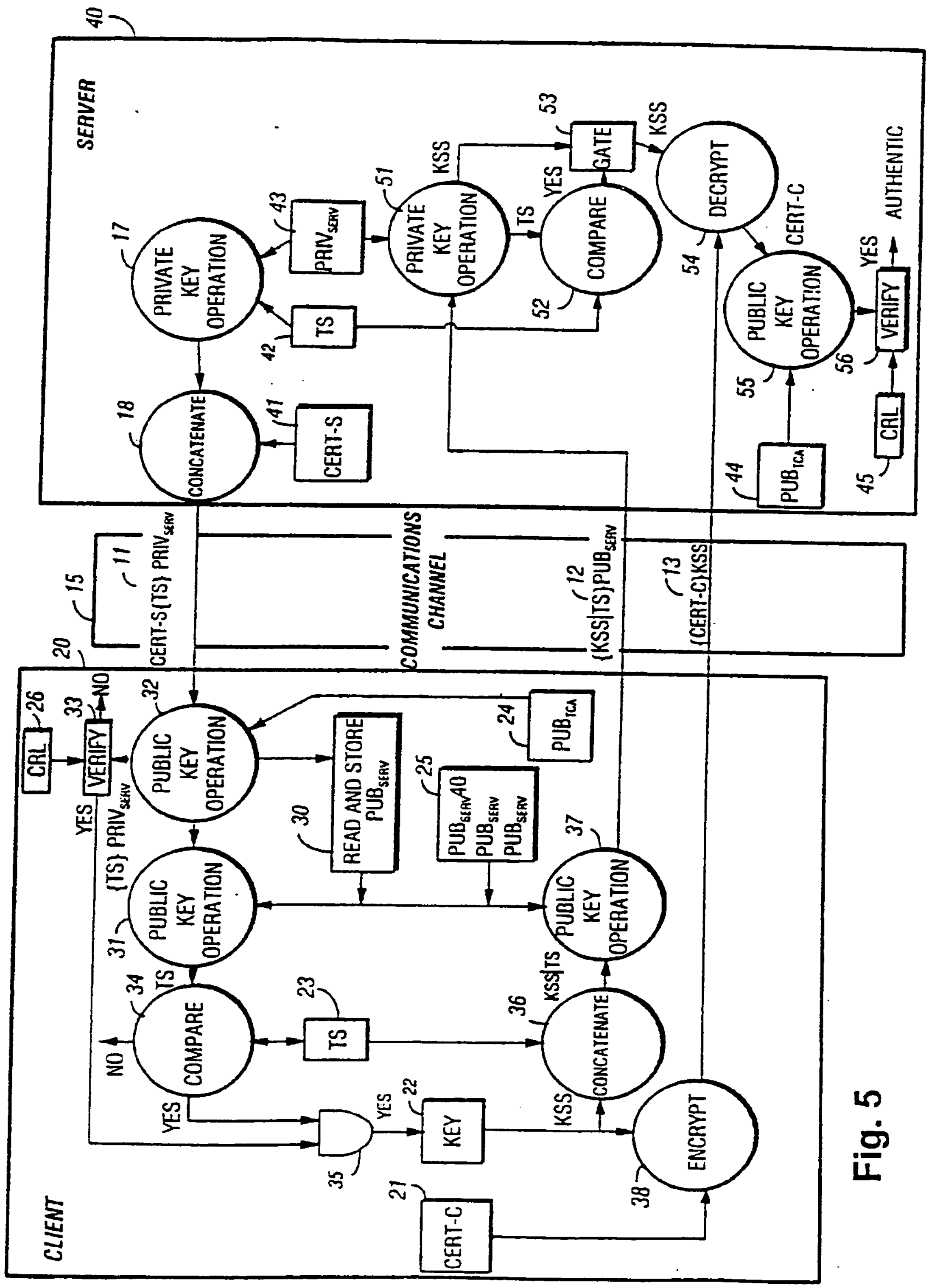

SERVER
PRIVATE KEY OPERATION
CONCATENATE
CERT-S
TS
PRIV$_{SERV}$
PRIVATE KEY OPERATION
KSS
COMPARE
YES
GATE
KSS
DECRYPT
CERT-C
PUBLIC KEY OPERATION
VERIFY
YES
AUTHENTIC
PUB$_{TCA}$
CRL
COMMUNICATIONS CHANNEL
CERT-S{TS} PRIV$_{SERV}$
{KSS|TS}PUB$_{SERV}$
{CERT-C}KSS
CLIENT
CRL
VERIFY
NO
YES
{TS} PRIV$_{SERV}$
PUBLIC KEY OPERATION
READ AND STORE PUB$_{SERV}$
PUB$_{TCA}$
PUB$_{SERV}$40
PUB$_{SERV}$
PUB$_{SERV}$
PUBLIC KEY OPERATION
PUBLIC KEY OPERATION
TS
KSS|TS
NO
COMPARE
TS
CONCATENATE
YES
YES
KEY
KSS
ENCRYPT
CERT-C

Fig. 5

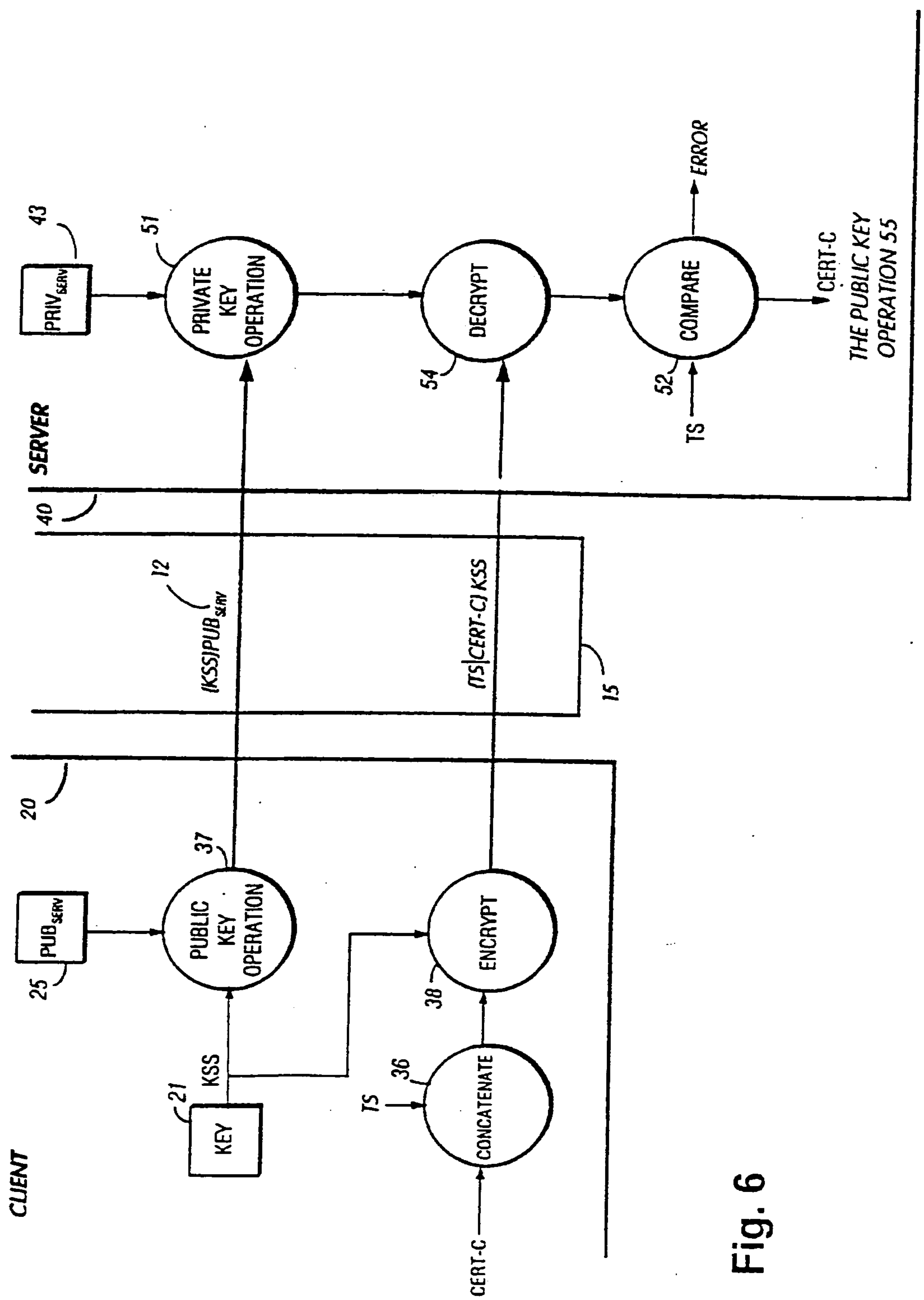

CLIENT
SERVER
KEY
21
KSS
PUBSERV
25
PUBLIC KEY OPERATION
37
20
CONCATENATE
36
TS
CERT-C
ENCRYPT
38
40
12
(KSS)PUBSERV
(TS|CERT-C) KSS
15
PRIVSERV
43
PRIVATE KEY OPERATION
51
DECRYPT
54
COMPARE
52
TS
ERROR
CERT-C
THE PUBLIC KEY OPERATION 55

Fig. 6

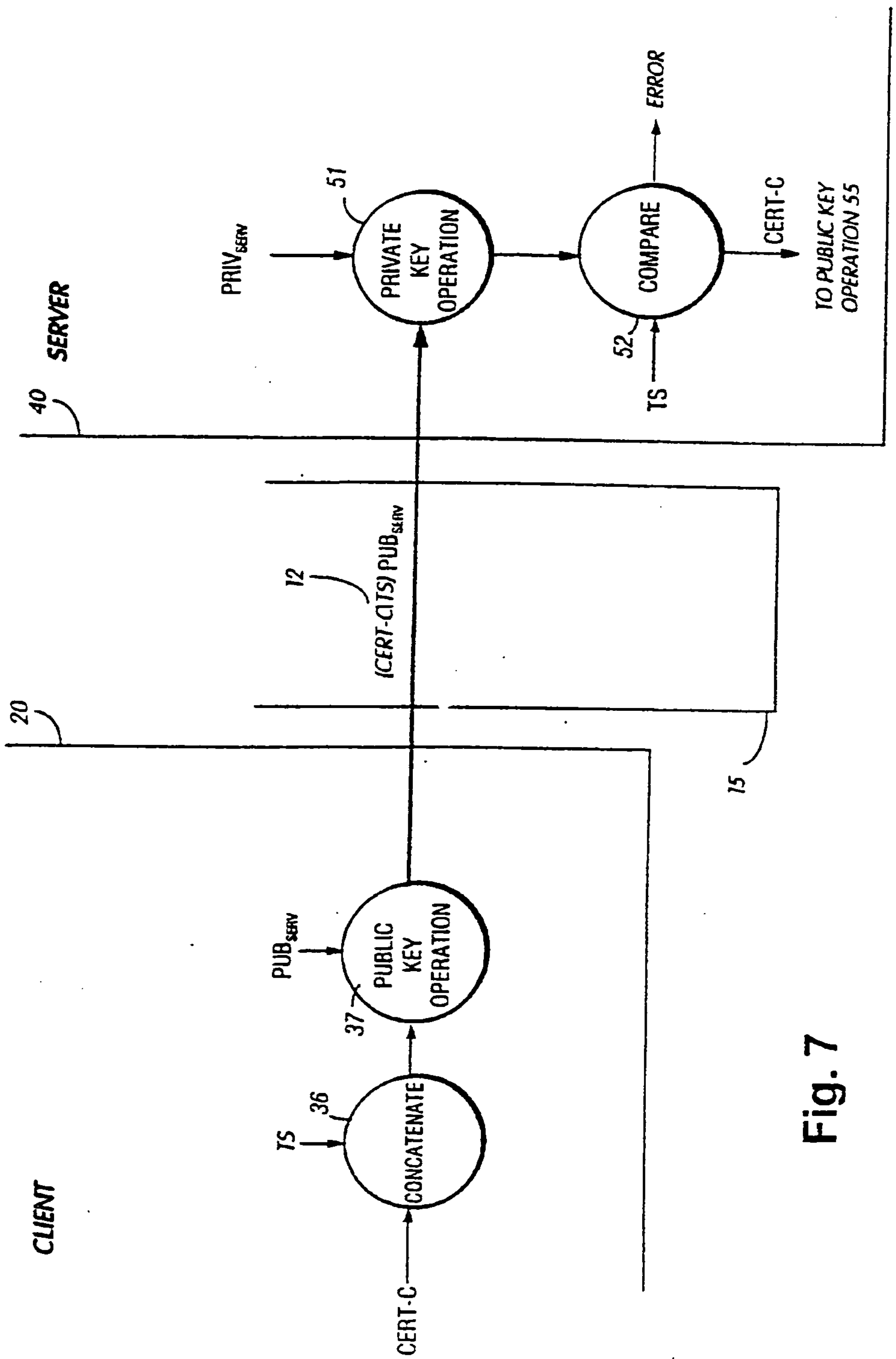

CLIENT
SERVER
CERT-C
TS
36
CONCATENATE
PUB$_{SERV}$
37
PUBLIC KEY OPERATION
20
12
(CERT-C|TS) PUB$_{SERV}$
15
40
PRIV$_{SERV}$
51
PRIVATE KEY OPERATION
COMPARE
52
TS
ERROR
CERT-C
TO PUBLIC KEY OPERATION 55

Fig. 7

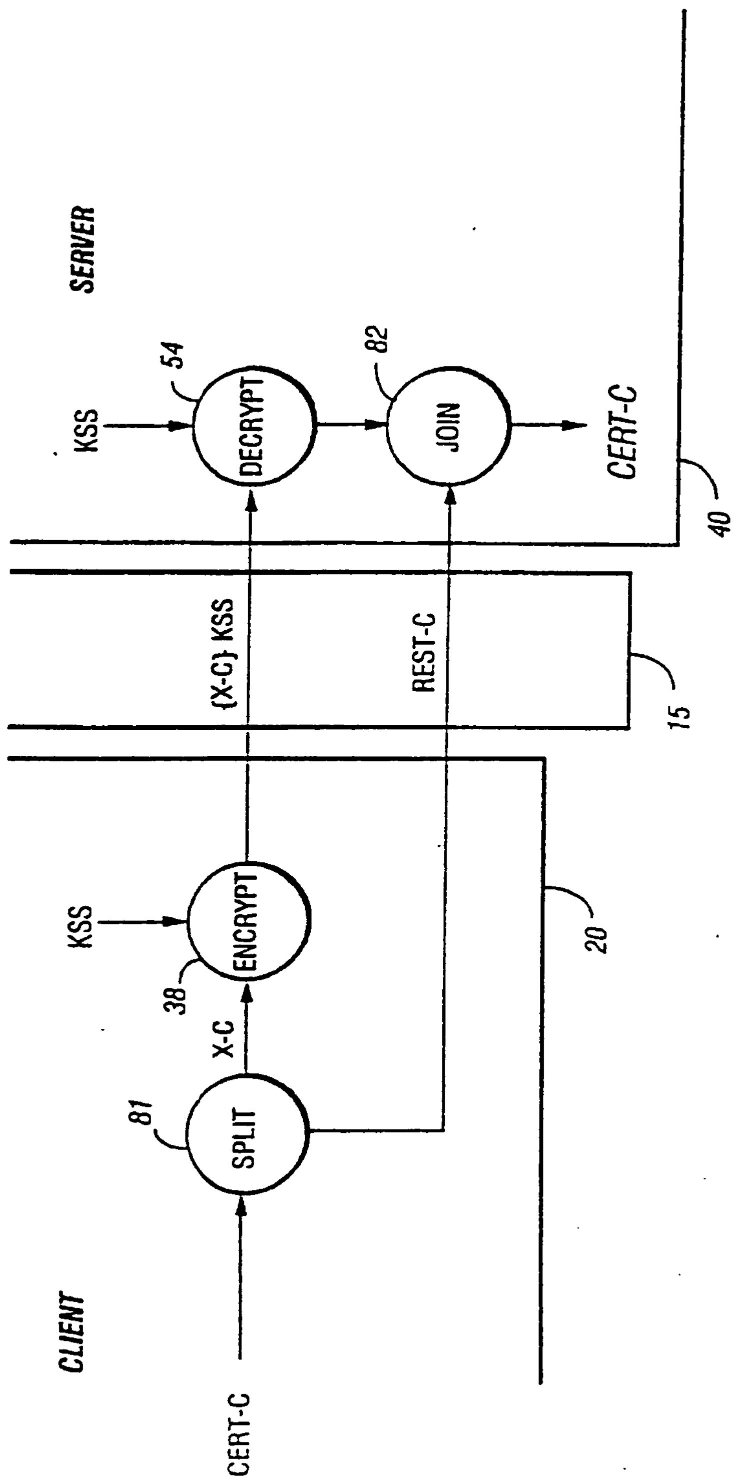

CLIENT
SERVER
CERT-C
SPLIT
81
X-C
ENCRYPT
38
KSS
KSS
DECRYPT
54
{X-C} KSS
REST-C
JOIN
82
CERT-C
15
20
40

Fig. 8

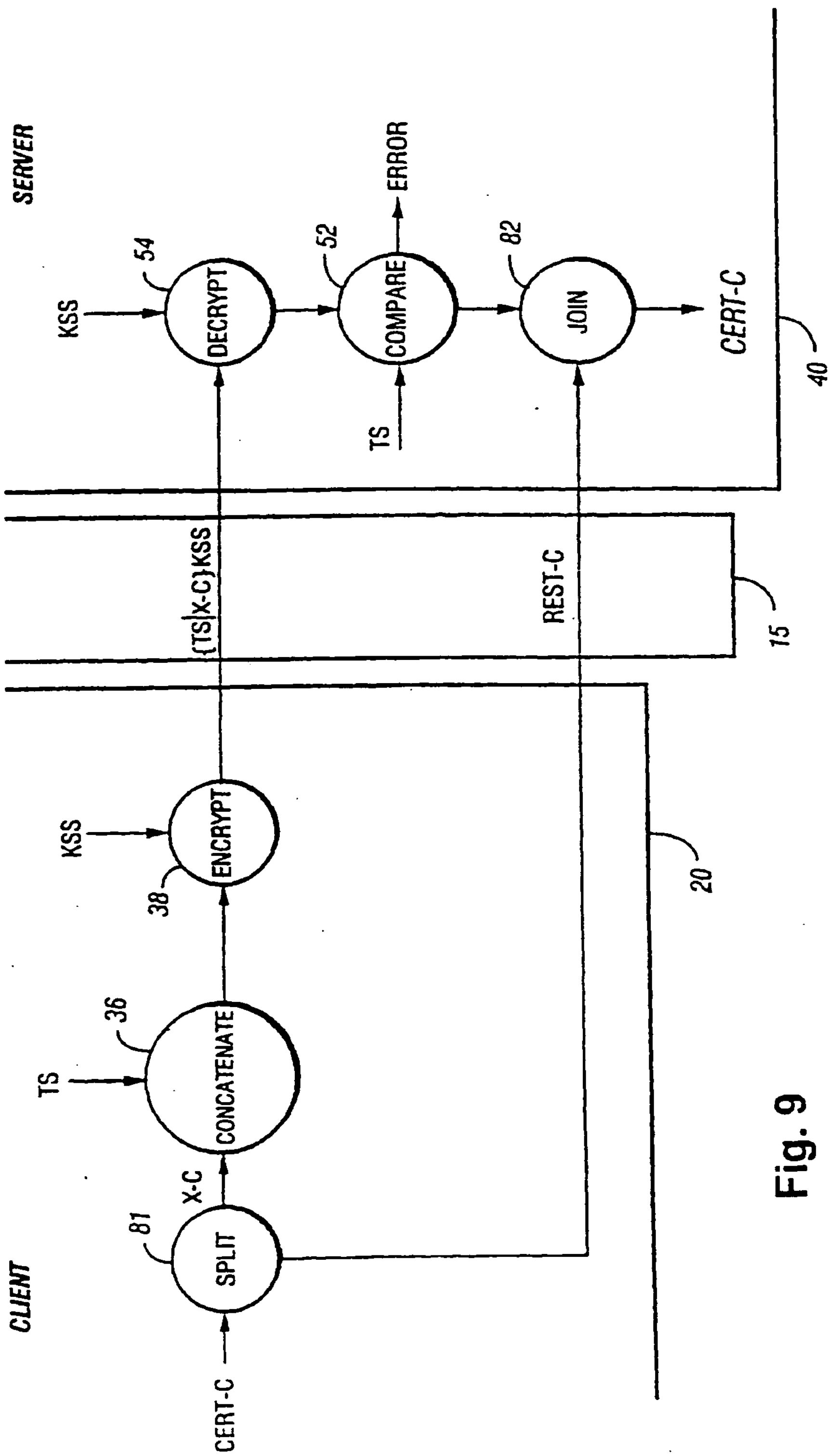

CLIENT
SERVER
CERT-C
SPLIT
81
X-C
TS
CONCATENATE
36
KSS
ENCRYPT
38
{TS|X-C}KSS
REST-C
15
20
KSS
DECRYPT
54
COMPARE
52
TS
ERROR
JOIN
82
CERT-C
40

Fig. 9

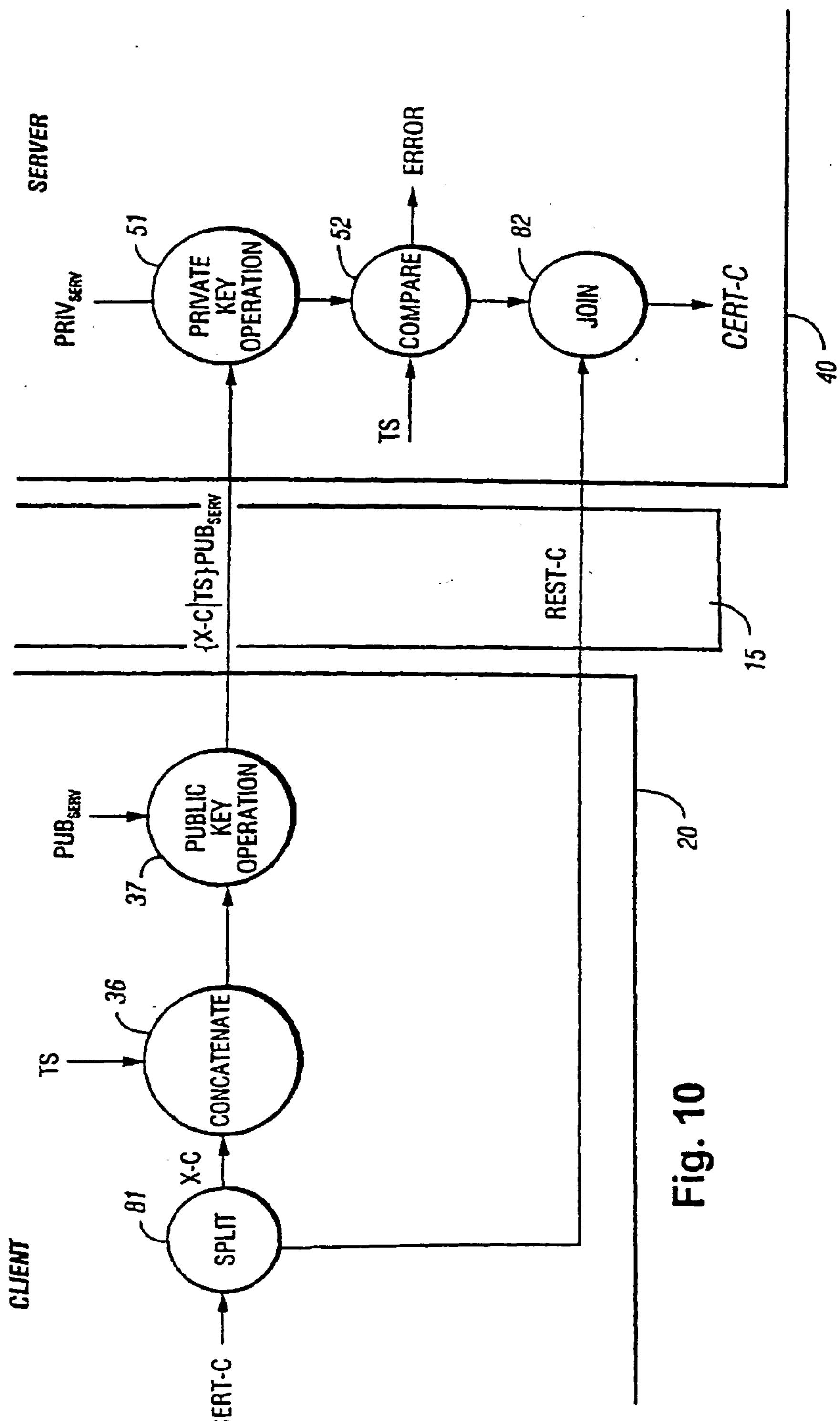

CLIENT
SERVER
CERT-C
SPLIT
81
X-C
TS
CONCATENATE
36
PUB_SERV
PUBLIC KEY OPERATION
37
{X-C|TS}PUB_SERV
REST-C
15
20
PRIV_SERV
PRIVATE KEY OPERATION
51
COMPARE
52
TS
ERROR
JOIN
82
CERT-C
40

Fig. 10

Fig. 11

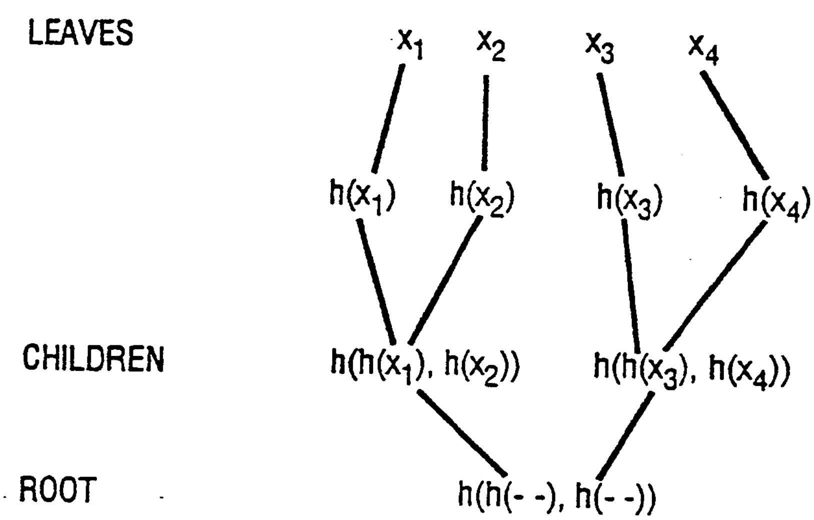
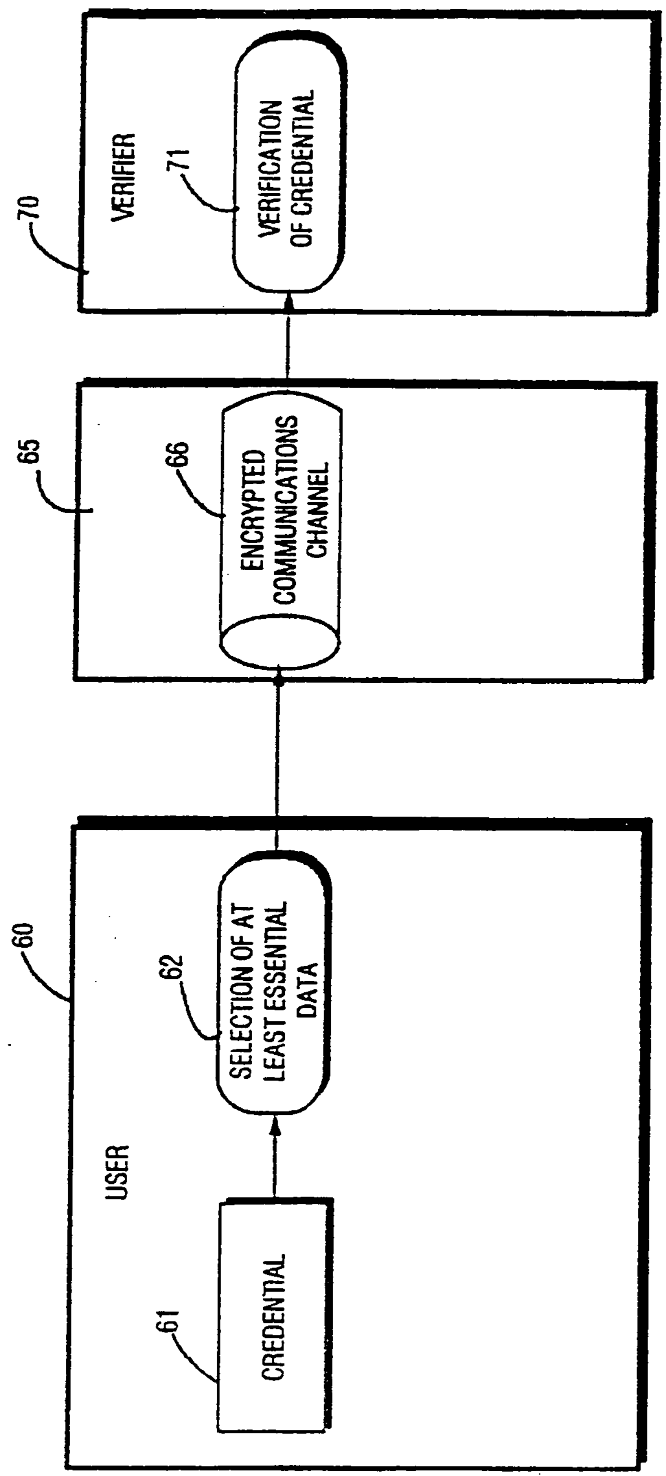

LEAVES
x_1 x_2 x_3 x_4
$h(x_1)$ $h(x_2)$ $h(x_3)$ $h(x_4)$
CHILDREN
$h(h(x_1), h(x_2))$ $h(h(x_3), h(x_4))$
ROOT
$h(h(- -), h(- -))$

Fig. 12



USER
60
61
CREDENTIAL
62
SELECTION OF AT LEAST ESSENTIAL DATA
65
66
ENCRYPTED COMMUNICATIONS CHANNEL
VERIFIER
70
71
VERIFICATION OF CREDENTIAL